# ShopDZ B2C 商城安装手册

北京康创联盛科技有限公司

2016年9月

1. **系统介绍**

ShopDZ B2C 商城是 ShopDZ于 2016 年最新推出，采用 PHP5+MySQL 技术为基础，采用 OOP（面向 对象）方式进行核心框架搭建，结合 MVC 模式进行开发的一套电商平台系统。

1. **运行环境**

|  |  |
| --- | --- |
| 项目 | 说明 |
| 操作系统 | 推荐使用 Linux，兼容 Windows，内存 >= 8G，硬盘 >= 300G |
| WEB服务器 | Apache 或 Nginx |
| 数据库 | MySQL5.5 及以上版本 |
| PHP | PHP5.3 及以上版本，支持以下扩展：  php\_mysql  php\_mysqli  php\_gd2  php\_mbstring  openssl  支持以下函数：  mysql\_connect,fsockopen,gethostbyname,file\_get\_contents,mb\_con vert\_encoding,json\_encode  其它项：  file\_uploads = On  upload\_max\_filesize = 1M |

推荐系统运行于 Linux + Nginx 环境。

1. **安装详解**

系统运行的最佳环境是 Linux + Nginx，以下以 centos 6.5\_X64 环境为例，介绍搭建过程。 提示：3.1 ~ 3.5 节是必需的安装步骤，从 3.6 节以后为可选配置。 首先需要安装依赖库。

首先需要安装依赖库。

yum -y install gcc gcc-c++ autoconf libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel ncurses ncurses-devel curl curl-devel e2fsprogs e2fsprogs-devel krb5 krb5-devel libidn libidn-devel openssl openssl-devel openldap openldap-devel nss\_ldap openldap-clients openldap-servers gd-devel libjpeg-devel libpng-devel freetype-devel libxml2-devel curl-devel freetype-devel bison gcc gcc-c++ autoconf automake zlib\* libxml\* ncurses-devel libtool-ltdl-devel\* pcre-devel zlib-devel gcc-g77 autoconf automake zlib\* fiex\* libxml\* ncurses-devel libmcrypt\* libtool-ltdl-devel\* perl gettext pcre-devel zlib-devel libXpm-devel libXpm icu icu4j libicu\* wget lrzsz mysql-devel unzip libaio rsync openssh-clients ntp

* 1. **安装 Nginx**

主配置文件：/usr/local/nginx/conf/nginx.conf

服务控制：service nginx (start|stop|restart|reload|configtest)

核心命令：

**# 下载依赖包**

cd /usr/local/src

wget <http://www.openssl.org/source/openssl-1.0.0l.tar.gz>

tar zxf openssl-1.0.0l.tar.gz

mv openssl-1.0.0l openssl

wget <http://zlib.net/zlib-1.2.8.tar.gz>

tar zxf zlib-1.2.8.tar.gz

mv zlib-1.2.8 zlib

wget <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.33.tar.gz>

tar zxf pcre-8.33.tar.gz

mv pcre-8.33 pcre

**# 安装**

nginx groupadd www

useradd -r -g www www -s /sbin/nologin

cd /usr/local/src/

wget <http://nginx.org/download/nginx-1.4.7.tar.gz>

tar zxf nginx-1.4.7.tar.gz

cd nginx-1.4.7

./configure \

"--prefix=/usr/local/nginx" \

"--sbin-path=/usr/local/nginx/sbin/nginx" \

"--conf-path=/usr/local/nginx/conf/nginx.conf" \

"--error-log-path=/usr/local/nginx/log/error.log" \

"--http-log-path=/usr/local/nginx/log/access.log" \

"--pid-path=/usr/local/nginx/var/nginx.pid" \

"--lock-path=/usr/local/nginx/var/nginx.lock" \

"--http-client-body-temp-path=/tmp/clientbody" \

"--http-proxy-temp-path=/tmp/proxy" \

"--http-fastcgi-temp-path=/tmp/fastcgi" \

"--http-uwsgi-temp-path=/tmp/uwsgi" \

"--http-scgi-temp-path=/tmp/scgi" \

"--user=www" \

"--group=www" \

"--with-file-aio" \

"--with-http\_realip\_module" \

"--with-http\_ssl\_module" \

"--with-openssl=/usr/local/src/openssl" \

"--with-http\_gzip\_static\_module" \

"--with-zlib=/usr/local/src/zlib" \

"--with-http\_stub\_status\_module" \

"--with-pcre=/usr/local/src/pcre" \

"--without-select\_module" \

"--without-poll\_module" \

"--without-http\_ssi\_module" \

"--without-http\_userid\_module" \

"--without-http\_geo\_module" \

"--without-http\_empty\_gif\_module" \

"--without-http\_map\_module" \

"--without-mail\_pop3\_module" \

"--without-mail\_imap\_module" \

"--without-mail\_smtp\_module"

make

make install

**# 注意：防火墙开放 80 端口**

**# 创建相关目录**

mkdir /usr/local/nginx/conf/vhosts

mkdir -p /var/log/nginx/shop

mkdir /var/run/nginx

chown -R www:www /var/log/nginx

chown -R www:www /var/run/nginx

mkdir /wwwroot

**主配置文件（/usr/local/nginx/conf/nginx.conf）内容如下:**

user www;

worker\_processes 2;

error\_log /var/log/nginx/error.log error;

pid /var/run/nginx/nginx.pid;

events {

worker\_connections 10240;

use epoll;

multi\_accept on;

}

http {

include mime.types;

default\_type application/octet-stream;

client\_body\_buffer\_size 8k;

client\_body\_temp\_path /tmp/client\_body\_temp;

client\_body\_timeout 30;

client\_header\_buffer\_size 32k;

large\_client\_header\_buffers 4 32k;

client\_header\_timeout 30;

client\_max\_body\_size 32m;

#keepalive\_disable msie6 safari;

keepalive\_timeout 3;

tcp\_nodelay on;

send\_timeout 30;

sendfile on;

tcp\_nopush on;

server\_names\_hash\_max\_size 512;

server\_names\_hash\_bucket\_size 128;

server\_tokens off;

open\_file\_cache off;

#index

index index.php index.html index.htm;

#fastcgi

fastcgi\_connect\_timeout 60;

fastcgi\_read\_timeout 60;

fastcgi\_send\_timeout 60;

fastcgi\_temp\_path /tmp/fastcgi\_temp 1 2;

fastcgi\_buffer\_size 64k;

fastcgi\_buffers 256 4k;

fastcgi\_max\_temp\_file\_size 256k;

fastcgi\_intercept\_errors on;

fastcgi\_index index.php;

#proxy

proxy\_temp\_path /tmp/proxy\_temp;

proxy\_buffer\_size 4k;

proxy\_buffering on;

proxy\_buffers 256 4k;

proxy\_busy\_buffers\_size 8k;

#gzip

gzip on;

gzip\_buffers 16 4k;

gzip\_comp\_level 1;

gzip\_http\_version 1.1;

gzip\_min\_length 1024;

gzip\_types text/css text/xml text/plain text/vnd.wap.wml application/x-javascript application/rss+xml application/xhtml+xml;

#realip module

set\_real\_ip\_from 127.0.0.1;

set\_real\_ip\_from 10.0.0.0/8;

set\_real\_ip\_from 172.16.0.0/20;

set\_real\_ip\_from 192.168.0.0/16;

set\_real\_ip\_from 192.168.1.0/24;

real\_ip\_header X-Real-IP;

#real\_ip\_header X-Forwarded-For;

#log

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

#virtualhost

include vhosts/\*;

}

**商城配置文件（/usr/local/nginx/conf/vhost/shop.conf）内容如下:**

server {

listen 0;

server\_name localhost;

access\_log /var/log/nginx/shop/access.log main;

root /wwwroot;

location ~ .\*\.(gif|jpg|jpeg|png|bmp|zip|exe|txt|ico|rar|htm|html)$

{

expires 30d;

}

location ~ .\*\.(swf|mp3|wmv|wma|mp4|mpg|flv)$

{

expires 30d;

}

location ~.\*(Runtime|Public|html|cache|diy|data|template|attachments |attachment|images|customavatars)/.\*\.php$ {

deny all;

}

location / {

index index.shtml index.html index.htm index.php;

}

location ~\* \.(tar|svn|tar.gz|zip)$ {

return 403;

}

location ~ ^/plugins/group/view/.\*\.(php|php5)$ {

return 403;

}

location ~ \.php {

rewrite ^/(.\*)/$ /$1 permanent;

fastcgi\_pass unix:/dev/shm/php-fcgi.sock;

fastcgi\_index index.php;

set $path\_info "";

set $real\_script\_name $fastcgi\_script\_name;

if ($fastcgi\_script\_name ~ "^(.+?\.php)(/.+)$") {

set $real\_script\_name $1;

set $path\_info $2;

}

fastcgi\_param SCRIPT\_FILENAME $document\_root$real\_script\_name;

fastcgi\_param SCRIPT\_NAME $real\_script\_name;

fastcgi\_param PATH\_INFO $path\_info;

include fastcgi\_params;

}

**服务脚本文件（/etc/init.d/nginx）内容如下：**

#!/bin/sh

#

# nginx - this script starts and stops the nginx daemon

#

# chkconfig: - 85 15

# description: Nginx is an HTTP(S) server, HTTP(S) reverse \

# proxy and IMAP/POP3 proxy server

# processname: nginx

# config: /usr/local/nginx/conf/nginx.conf

# config: /usr/local/nginx/sysconfig/nginx

# pidfile: /usr/local/nginx/var/nginx.pid

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/local/nginx/sbin/nginx"

prog=$(basename $nginx)

sysconfig="/usr/local/nginx/sysconfig/$prog"

lockfile="/usr/local/nginx/lock"

pidfile="/usr/local/nginx/var/${prog}.pid"

NGINX\_CONF\_FILE="/usr/local/nginx/conf/nginx.conf"

[ -f $sysconfig ] && . $sysconfig

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo [ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc -p $pidfile $prog

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

configtest\_q || return 6

stop

start

}

reload() {

configtest\_q || return 6

echo -n $"Reloading $prog: "

killproc -p $pidfile $prog -HUP

echo

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

configtest\_q() {

$nginx -t -q -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

# Upgrade the binary with no downtime.

upgrade() {

local oldbin\_pidfile="${pidfile}.oldbin"

configtest\_q || return 6

echo -n $"Upgrading $prog: "

killproc -p $pidfile $prog -USR2

retval=$?

sleep 1

if [[ -f ${oldbin\_pidfile} && -f ${pidfile} ]]; then

killproc -p $oldbin\_pidfile $prog -QUIT

success $"$prog online upgrade"

echo

return 0

else

failure $"$prog online upgrade"

echo

return 1

fi

}

# Tell nginx to reopen logs

reopen\_logs() {

configtest\_q || return 6

echo -n $"Reopening $prog logs: "

killproc -p $pidfile $prog -USR1

retval=$?

echo

return $retval

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest|reopen\_logs)

$1

;;

force-reload|upgrade)

rh\_status\_q || exit 7

upgrade

;;

reload)

rh\_status\_q || exit 7

$1

;;

status|status\_q)

rh\_$1

;;

condrestart|try-restart)

rh\_status\_q || exit 7

restart

;;

\*)

echo $"Usage: $0 {start|stop|reload|configtest|status|force-reload|upgrade|restart|reopen\_logs}"

exit 2

esac

**添加到系统服务并开机启动**

chmod u+x /etc/init.d/nginx

chkconfig --add nginx

chkconfig nginx on

**3.2 安装 PHP**

**3.2.1 安装 PHP 基础环境**

PHP 配置文件：/usr/local/php/etc/php.ini

PHP-FPM 配置文件：/usr/local/php/etc/php-fpm.conf

服务控制：service php-fpm (start|stop|restart|reload)

核心命令：

# 安装依赖库

cd /usr/local/src

wget <ftp://mcrypt.hellug.gr/pub/crypto/mcrypt/libmcrypt/libmcrypt-2.5.7.tar.gz>

tar zxvf libmcrypt-2.5.7.tar.gz

cd libmcrypt-2.5.7/

./configure

make

make install

/sbin/ldconfig

cd libltdl/

./configure --enable-ltdl-install

make

make install

yum install mysql-devel

ln -s /usr/lib64/libXpm.so\* /usr/lib/

mkdir /usr/lib/mysql

ln -s /usr/lib64/mysql/libmysqlclient\* /usr/lib/mysql

# 安装 PHP

wget <http://tw1.php.net/get/php-5.4.26.tar.gz/from/this/mirror>

tar zxf php-5.4.26.tar.gz

cd php-5.4.26

./configure \

--prefix=/usr/local/php \

--with-config-file-path=/usr/local/php/etc \

--with-mysql=mysqlnd \

--with-mysqli=mysqlnd \

--with-pdo-mysql=mysqlnd \

--with-libxml-dir \

--with-pcre-dir \

--with-mcrypt \

--with-openssl \

--with-gd \

--with-jpeg-dir \

--with-png-dir \

--with-xpm-dir \

--with-gettext \

--with-freetype-dir \

--with-bz2 \

--with-iconv-dir \

--with-zlib \

--with-zlib-dir \

--with-curl \

--with-xmlrpc \

--enable-fpm \

--with-fpm-user=www \

--with-fpm-group=www \

--with-icu-dir=/usr \

--enable-pcntl \

--enable-intl \

--enable-sockets \

--enable-gd-native-ttf \

--enable-gd-jis-conv \

--enable-ftp \

--enable-zip \

--enable-calendar \

--enable-mbstring \

--enable-exif \

--enable-soap \

--enable-wddx \

--enable-bcmath \

--enable-fd-setsize=4096 \

--disable-short-tags \

--disable-ipv6

make

make install

# 创建配置文件

cp /usr/local/src/php-5.4.26/php.ini-production /usr/local/php/etc/php.ini

cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf

# 安装 zendloader 扩展

wget <http://downloads.zend.com/guard/6.0.0/ZendGuardLoader-70429-PHP-5.4-linux-glibc23-x86_64.tar.gz>

tar zxf ZendGuardLoader-70429-PHP-5.4-linux-glibc23-x86\_64.tar.gz

cd ZendGuardLoader-70429-PHP-5.4-linux-glibc23-x86\_64

cd php-5.4.x/

mv ZendGuardLoader.so /usr/local/php/lib/php/extensions/no-debug-non-zts-20100525/ZendGuardLoader.so

vi /usr/local/php/etc/php.ini

# 追加以下内容

[zendloader]

zend\_loader.enable=1

zend\_extension=/shop\_app/php/lib/php/extensions/no-debug-non-zts-20100525/ZendGuardLoader.so

zend\_loader.disable\_licensing=0

zend\_loader.obfuscation\_level\_support=3

zend\_loader.license\_path=

设置 php-fpm 配置,修改以下参数项。

vi /usr/local/php/etc/php-fpm

pid = /var/run/php/php-fpm.pid

error\_log = /var/log/php/error.log

user = www

group = www

listen = 127.0.0.1:9000

pm = static

# php-fpm 加入系统服务并开机启动

cp /usr/local/src/php-5.4.26/sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm

# 编辑 PHP 相关路径(/etc/init.d/php-fpm)

chmod u+x /etc/init.d/php-fpm

chkconfig --add php-fpm

service php-fpm start

chkconfig php-fpm on

**3.3 安装 MySQL**

主配置文件：/usr/local/mysql/my.cnf

数据库目录：/usr/local/mysql/data

服务器控制：service mysqld (start|stop|restart)

核心命令：

cd /usr/local/src

wget <http://cdn.mysql.com/Downloads/MySQL-5.6/mysql-5.6.17-linux-glibc2.5-x86_64.tar.gz>

groupadd mysql

useradd -r -g mysql mysql -s /sbin/nologin

./scripts/mysql\_install\_db --user=mysql --datadir=/var/lib/mysql/

vi /usr/local/mysql/my.cnf

[mysqld]

basedir = /usr/local/mysql

datadir = /var/lib/mysql

port = 3306

server\_id = 1

socket = /var/lib/mysql/mysql.sock

log\_error = /var/log/mysql/error.log

log-bin = /var/log/mysql/log\_bin/log\_bin

binlog-do-db=shop

sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES

[mysqld\_safe]

pid-file = /var/run/mysql/mysqld.pid

mkdir /var/lib/mysql

mkdir /var/log/mysql

mkdir /var/log/mysql/log\_bin

mkdir /var/run/mysql

chown -R mysql:mysql /var/lib/mysql/

chown -R mysql:mysql /var/log/mysql/

chown -R mysql:mysql /var/run/mysql/

# 测试启动

bin/mysqld\_safe --user=mysql &

vi support-files/mysql.server

basedir=/usr/local/mysql

datadir=/var/lib/mysql

mysqld\_pid\_file\_path=/var/run/mysql/mysqld.pid

# 加入系统服务

cp support-files/mysql.server /etc/init.d/mysqld

chkconfig --add mysqld

chkconfig mysqld on

service mysqld start

# 设置 ROOT 密码

./bin/mysqladmin -u root password 'xxxxxx'

**3.4安装商城程序**

在 Nginx、PHP、MySQL 安装并配置完成以后，可以安装 ShopDZ B2C 商城了。

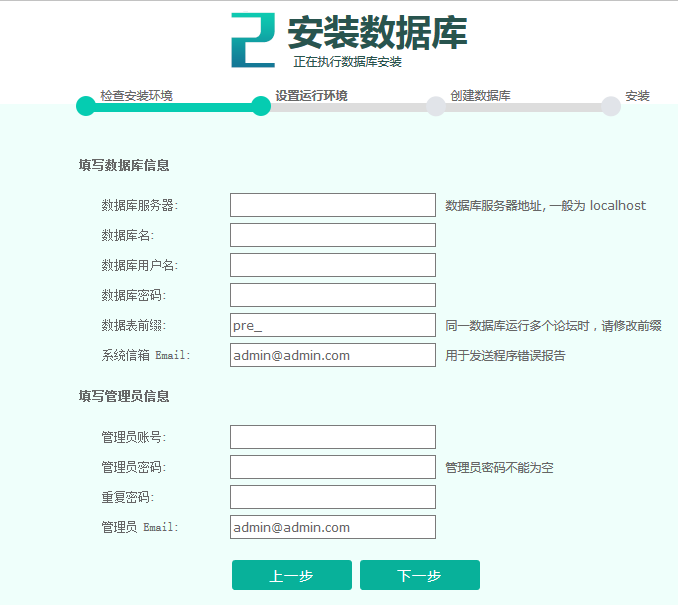
首先需要仔细阅读安装协议



阅读并同意后开始检查安装环境，如果检测有未通过项，将显示红色提示并无法进行下一步



待所有检查项均通过后，进入下一步选择，输入数据库账号等信息



下一步开始安装数据库



安装数据库成功，下一步提示安装成功并显示安装相关信息



安装完成后，进入后台 》店铺 》清理缓存。



点击后台右上解的商城首页链接，查看一下商城首页，内容正确展示，至此安装过程顺利完成。



**3.5 设置定时任务**

以 Linux 下为例设定时任务。

# /usr/local/php/bin/php 为 php 可执行文件所在路径

# 按分钟执行的任务

# 执行频率越高用户体验超高， 但系统压力也越大，建议执行频率 10 分钟。

/usr/local/php/bin/php /wwwroot/crontab.php Minutes

# 按小时执行的任务

/usr/local/php/bin/php /wwwroot/crontab.php Hour

# 按天执行的任务

/usr/local/php/bin/php /wwwroot/crontab.php Date

# 按月执行的任务

/usr/local/php/bin/php /wwwroot/crontab.php Month

示例：

# 每 10 分钟执行一次

\*/10 \* \* \* \* /usr/local/php/bin/php /wwwroot/crontab.php minutes

# 每 1 小时执行一次

10 \* \* \* \* /usr/local/php/bin/php /wwwroot/crontab.php hour

# 每天 3:20 执行一次

20 3 \* \* \* /usr/local/php/bin/php /wwwroot/crontab.php date

# 每月 1 号 3:30 执行

30 3 1 \* \* /usr/local/php/bin/php /wwwroot/crontab.php month